IMPLEMENTATION OF THE BLOCK-KRYLOV BOUNDARY FLEXIBILITY METHOD OF COMPONENT SYNTHESIS

Kelly S.Carney
National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio 44135

Ayman A. Abdallah
Analex Corporation
3001 Aerospace Parkway
Brook Park, Ohio 44142

Arthur A. Hucklebridge
Case Western Reserve University
Department of Civil Engineering
Cleveland, Ohio 44106

Abstract

A method of dynamic substructuring is presented which utilizes a set of static Ritz vectors as a replacement for normal eigenvectors in component mode synthesis. This set of Ritz vectors is generated in a recurrence relationship, which has the form of a block-Krylov subspace. The initial seed to the recurrence algorithm is based on the boundary flexibility vectors of the component. This algorithm is not load-dependent, is applicable to both fixed and free-interface boundary components, and results in a general component model appropriate for any type of dynamic analysis. This methodology has been implemented in the MSC/NASTRAN normal modes solution sequence using DMAP. The accuracy is found to be comparable to that of component synthesis based upon normal modes. The block-Krylov recurrence algorithm is a series of static solutions and so requires significantly less computation than solving the normal eigenspace problem.

1993 MSC World Users' Conference
Arlington, Virginia
May 24-28, 1993

IMPLEMENTATION OF THE BLOCK-KRYLOV BOUNDARY FLEXIBILITY METHOD OF COMPONENT SYNTHESIS

Kelly S.Carney
National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio 44135

Ayman A. Abdallah
Analex Corporation
3001 Aerospace Parkway
Brook Park, Ohio 44142

Arthur A. Hucklebridge
Case Western Reserve University
Department of Civil Engineering
Cleveland, Ohio 44106

Introduction

Component mode synthesis is a methodology for analyzing large structures by separating them into smaller components, which can then be recombined to analyze the entire system. This methodology has become well established and widely used in structural dynamic analysis. The advantages of component mode synthesis include, the lower computation costs associated with the smaller components which are analyzed, and the flexibility of data management gained by working with the discrete components.

The typical component mode synthesis algorithm is briefly described [1]. A large structure is broken into components, with each component having a set of boundary, or interface, points. At these interface points, fixed or free boundary conditions are assumed, and a corresponding set of component normal mode shapes, or eigenvectors, is determined. The eigenvectors are augmented by a set of modes which are associated with the component's boundary flexibility. Depending on whether a fixed or free interface is selected, these modes are the constraint modes or the attachment modes, respectively. The combined set of component normal modes and boundary modes are used to represent the component in subsequent system analysis, by using the following transformation process. The combined set of modes form a coordinate transformation matrix which transforms the physical coordinates of the structural model into a combination of modal coordinates and boundary coordinates. The boundary coordinates are retained in the physical space, so they can be used to couple the components for subsequent system analysis.

A component's size, although smaller than that of the complete structural model, can still be large enough to make computation expensive. The rapid reduction in cost per calculation in today's digital computers has not necessarily led to a reduction in total computation cost. Instead, engineers have exploited the increased computational resources by creating larger structural and component models. The larger models have allowed for more structural details to be included, as well as more refined data recovery, but they may be expensive to formulate and analyze. In order to reduce the computational cost associated with large component models, it is desirable to develop more efficient methods of formulation. Since the solution of the normal eigensystem problem requires the largest computational effort in component formulation, it is logical to develop alternate methods which circumvent the eigensystem solution entirely.

A method, which does circumvent the eigensystem solution, has been defined in literature and is briefly described [2-4]. The boundary flexibility modes, specifically either the same constraint modes or attachment modes that were mentioned previously, are multiplied by the component mass matrix to create a force matrix.

Static analysis is then performed, using this force matrix and the component stiffness matrix, to obtain a matrix, or block, of vector displacements. A recurrence relationship of matrix multiplications, which have been shown to be a Krylov sequence [3,5], then defines a series of matrices, or blocks, of vector displacements. The calculated vectors are orthogonalized, using normalized Gram-Schmidt orthogonalization [6]. These vectors, which can also be thought of as static modes, replace the normal modes in the component formulation methodology. Because the vectors are calculated in blocks and are based on a Krylov sequence, the subspace defined by these vectors is called a block-Krylov subspace.

The primary contribution described in this paper is the incorporation of the above methodology into MSC/NASTRAN. Since the block-Krylov vectors replace the normal modes in typical component mode synthesis, the physical to modal coordinate transformation, the constraint modes, and the attachment modes, are not changed. Therefore, the component mode synthesis formulation currently available in MSC/NASTRAN was utilized, except for the substitution of the normal modes by the block-Krylov vectors. In addition to the contribution described above, the technique of using Gram-Schmidt orthogonalization within each vector block was added to the methodology.

Problem Definition and Theory

Overview

Wilson, Yuan, and Dickens [7] originally proposed the use of Ritz vectors, based upon external loading, for structural dynamic analysis. This formulation reduced an entire structure, not a component. The algorithm begins with a set of externally applied loads. The displacements from the static solution to the applied loads become the initial Ritz vector. That vector is then multiplied by the mass matrix to become the next force vector. This sequence is repeated to form a recurrence relationship. This recurrence relationship is used in the papers discussed below and throughout this work.

Nour-Omid and Clough [5] investigated Wilson, et al.'s methodology and found that the proposed recurrence relationship was actually a Krylov sequence. A Krylov subspace of order j is a vector space defined by

$$[\phi, \mathbf{A}\phi, \mathbf{A}^2\phi, \ldots, \mathbf{A}^{j-1}\phi] \tag{1}$$

where ϕ is a column vector and $\mathbf{A}$ is a square matrix. If $\mathbf{A}$ is $n \times n$ dimensional, and if $j = n$, the Krylov vectors span the n-dimensional space [3], and an exact solution can be produced. In structural dynamics, the Krylov subspace can be defined by the following sequence,

$$[\mathbf{r}, \mathbf{K}^{-1}\mathbf{M}\mathbf{r}, (\mathbf{K}^{-1}\mathbf{M})^2\mathbf{r}, \ldots, (\mathbf{K}^{-1}\mathbf{M})^{j-1}\mathbf{r}] \tag{2}$$

where $\mathbf{K}$ is the stiffness matrix, $\mathbf{M}$ is the mass matrix, and $\mathbf{r}$ is a starting vector (or in block-Krylov, a set of vectors). This Krylov sequence is identical to the Lanczos eigenvalue extraction algorithm, when applied with complete orthogonalization with respect to both the mass and the stiffness matrices.

The use of Krylov vectors was shown to be applicable to component mode synthesis by Wilson and Bayo [8]. The Ritz vectors calculated were based, once again, upon an external load. Only a formulation for components with fixed interface boundary conditions was presented.

Craig and Hale [3], and Abdallah and Hucklebridge [2], demonstrated a methodology applicable to components with fixed or free interfaces, with or without rigid body modes, and with no applied loading. Components having no applied external loading were formed using the boundary flexibility matrix, multiplied by the mass matrix, to form a force matrix. This force matrix produces a set of vectors, which are referred to as a block. As discussed previously, the boundary flexibility matrix is defined as either the constraint modes or the attachment modes, depending on whether fixed or free interface conditions are selected. The methodology contained in these two papers is reviewed in the next three parts of this section. Abdallah and Hucklebridge also quantified the advantages, in computational effort, that block-Krylov vectors have over normal eigenvectors.

Yiu and Landress [4] also developed a methodology for forming a component which does not have an external applied load. However, their formulation is applicable to fixed interface components only. A criteria for concluding the recurrence sequence, based upon the rigid body mass and flexibility represented by the calculated Ritz vectors, was proposed.

Fixed Interface Methodology

First, as is standard in component mode synthesis methods, the finite element component mass, $\mathbf{m}$, and stiffness, $\mathbf{k}$, matrices are partitioned into internal and external degrees of freedom, denoted by subscripts i and c, respectively.

$$\mathbf{m} = \begin{bmatrix} \mathbf{m}_{cc} & \mathbf{m}_{ci} \\ \mathbf{m}_{ic} & \mathbf{m}_{ii} \end{bmatrix} \tag{3}$$

$$\mathbf{k} = \begin{bmatrix} \mathbf{k}_{cc} & \mathbf{k}_{ci} \\ \mathbf{k}_{ic} & \mathbf{k}_{ii} \end{bmatrix} \tag{4}$$

The constraint modes are defined by

$$\mathbf{\Phi}_{ic} = -\mathbf{k}_{ii}^{-1}\,\mathbf{k}_{ic} \tag{5}$$

which is the same definition used in standard component mode synthesis.

For Wilson's method [7], a set of externally applied loads is required to obtain the initial set of Ritz vectors. For the boundary flexibility method, this set of loads is created by multiplying the constraint modes by the mass matrix. (Craig [3] also included the off-diagonal mass matrix in his formulation.) Since the mass matrix is used to create the loads, they can be considered inertia loads. This set of inertia loads are then used to generate the initial set, or block, of Ritz vectors using the following

$$\mathbf{q}_1^{**} = \mathbf{k}_{ii}^{-1}\left(\mathbf{m}_{ii}\mathbf{\Phi}_{ic} + \mathbf{m}_{ic}\right) \tag{6}$$

where the superscript ** indicates that the vectors in the matrix have not been normalized. The first block of vectors is normalized using the following equation. The subscript r, in the following equation, signifies that the block is normalized vector by vector, and there are c vectors within each matrix, or block.

$$\mathbf{q}_{1_r} = \frac{\mathbf{q}_{1_r}^{**}}{\sqrt{\left(\mathbf{q}_{1_r}^{**T}\,\mathbf{m}_{ii}\mathbf{q}_{1_r}^{**}\right)}} \qquad r = 1,2,\ldots,c \tag{7}$$

The subsequent sets of block-Krylov vectors are generated using the recurrence relationship [5,7], which was defined in Eq. (2),

$$\mathbf{q}_j^* = \mathbf{k}_{ii}^{-1}\,\mathbf{m}_{ii}\,\mathbf{q}_{j-1} \tag{8}$$

where the superscript * signifies that the vectors have not been orthogonalized or normalized. The additional sets of vectors are orthogonalized, with respect to the mass matrix, with all previous vectors. The process used to perform this orthogonalization is a normalized Gram-Schmidt procedure.

$$\mathbf{q}_j^{**} = \mathbf{q}_j^* - \mathbf{q}_{1,j-1}\mathbf{c} \tag{9}$$

where

$$\mathbf{c} = \mathbf{q}_{1,j-1}^{T}\,\mathbf{m}_{ii}\mathbf{q}_j^* \tag{10}$$

and $\mathbf{q}_{1,j\text{-}1}$ is the concatenation of the previous sets of block-Krylov vectors,

$$\mathbf{q}_{1,j-1} = [\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_{j-1}] \tag{11}$$

where all vectors have been normalized as follows.

$$\mathbf{q}_{j_r} = \frac{\mathbf{q}_{j_r}^{**}}{\sqrt{(\mathbf{q}_{j_r}^{**T}\,\mathbf{m}_{ii}\mathbf{q}_{j_r}^{**})}} \qquad r = 1,2,\ldots,c \tag{12}$$

The complete set of calculated block-Krylov Ritz vectors is included in the transformation matrix as $\mathbf{Q}_l$. (The resulting transformation matrix has the same form as that of "Craig-Bampton" component mode synthesis [9], with the Krylov vectors replacing the normal modes.)

$$\Psi = \begin{bmatrix} \mathbf{I}_{cc} & \mathbf{0} \\ \Phi_{ic} & \mathbf{Q}_l \end{bmatrix} \tag{13}$$

The physical mass and stiffness matrices are transformed into the component modal matrices

$$\mu = \Psi^T\,\mathbf{m}\,\Psi \tag{14}$$

$$\kappa = \Psi^T\,\mathbf{k}\,\Psi \tag{15}$$

The resulting mass submatrices are

$$\mu = \begin{bmatrix} \mu_{cc} & \mu_{cl} \\ \mu_{lc} & \mu_{ll} \end{bmatrix} \tag{16}$$

where

$$\mu_{cc} = \Phi_{ic}^{T}(\mathbf{m}_{ii}\Phi_{ic} + \mathbf{m}_{ic}) + \mathbf{m}_{ci}\Phi_{ic} + \mathbf{m}_{cc} \tag{17}$$

$$\mu_{lc} = \mu_{cl}^{T} = \mathbf{Q}_l^{T}(\mathbf{m}_{ii}\Phi_{ic} + \mathbf{m}_{ic}) \tag{18}$$

$$\mu_{ll} = \mathbf{I}_{ll} = \mathbf{Q}_l^T \mathbf{m}_{ii} \mathbf{Q}_l \tag{19}$$

The resulting stiffness submatrices are

$$\kappa = \begin{bmatrix} \kappa_{cc} & \kappa_{cl} \\ \kappa_{lc} & \kappa_{ll} \end{bmatrix} \tag{20}$$

where

$$\kappa_{cc} = \mathbf{k}_{ci} \Phi_{ic} + \mathbf{k}_{cc} \tag{21}$$

$$\kappa_{lc} = \kappa_{cl}^{\ T} = 0 \tag{22}$$

$$\kappa_{ll} = \mathbf{Q}_l^T \mathbf{k}_{ii} \mathbf{Q}_l \tag{23}$$

The use of constraint modes in the transformation matrix leads to the null off-diagonal partitions of the component stiffness matrix, just as in the approach based upon normal modes.

Free Interface Methodology for Components with No Rigid Body Modes

When allowing the interface points of a component to be free to deflect while forming the component, a somewhat different basis for the initial vector of the Krylov algorithm is required. The attachment modes, rather than the constraint modes are utilized in initial block definition. By definition, the attachment modes are the columns of the flexibility matrix which correspond to the interface degrees of freedom.

$$\mathbf{g} = \mathbf{k}^{-1} \tag{24}$$

$$\mathbf{g}_a = \begin{bmatrix} \mathbf{g}_{cc} \\ \mathbf{g}_{ic} \end{bmatrix} \tag{25}$$

The initial block of vectors in the free interface formulation is defined as

$$\mathbf{q}_1^{**} = \mathbf{k}^{-1} \mathbf{m}\, \mathbf{g}_a \tag{26}$$

and is normalized as follows.

$$\mathbf{q}_{1_r} = \frac{\mathbf{q}_{1_r}^{**}}{\sqrt{(\mathbf{q}_{1_r}^{**T}\ \mathbf{m}\ \mathbf{q}_{1_r}^{**})}} \qquad r = 1,2,\ldots,c \tag{27}$$

Note that the unpartitioned physical mass and stiffness matrices of the component are used in the free interface formulation. The recurrence algorithm then proceeds in the same manner as in the fixed interface methodology.

$$\mathbf{q}_j^{*} = \mathbf{k}^{-1}\mathbf{m}\,\mathbf{q}_{j-1} \tag{28}$$

$$\mathbf{q}_j^{**} = \mathbf{q}_j^{*} - \mathbf{q}_{1,j-1}\mathbf{c} \tag{29}$$

$$\mathbf{c} = \mathbf{q}_{1,j-1}^{T}\,\mathbf{m}\,\mathbf{q}_j^{*} \tag{30}$$

$$\mathbf{q}_{j_r} = \frac{\mathbf{q}_{j_r}^{**}}{\sqrt{(\mathbf{q}_{j_r}^{**T}\,\mathbf{m}\,\mathbf{q}_{j_r}^{**})}} \qquad r = 1,2,\ldots,c \tag{31}$$

Formation of block-Krylov component then follows the normal component mode synthesis techniques which were presented by MacNeal [10] and Rubin [11]. To combine the "Rubin-MacNeal" method with the presented method, the normal eigenvectors are simply replaced with the block-Krylov Ritz vectors, as in the fixed interface methodology. The free interface methodology uses residual flexibility terms, which fully define the stiffness missing from the modal space due to excluded modes and are described below. The flexibility contained in the calculated Krylov vectors is given by the following equation.

$$\mathbf{g}_k = \mathbf{Q}_l\,(\mathbf{Q}_l^{T}\mathbf{k}\,\mathbf{Q}_l)^{-1}\mathbf{Q}_l^{T} \tag{32}$$

The unrepresented flexibility, or residual flexibility, is defined as

$$\mathbf{g}_d = \mathbf{g} - \mathbf{g}_k \tag{33}$$

The residual flexibility matrix is then partitioned in the same manner as the flexibility matrix was in Eq. (25), when the attachment modes were created for initial vector calculation. The result is the residual attachment modes.

$$\mathbf{g}_{a_d} = \begin{bmatrix} \mathbf{g}_{cc_d} \\ \mathbf{g}_{ic_d} \end{bmatrix} \tag{34}$$

When the residual attachment modes, $\mathbf{g}_{ad}$, are added to the Krylov modes, $\mathbf{Q}_l$, the complete flexibility of the component is represented.

The residual attachment modes and the Krylov modes are used to form the component transformation matrix. This matrix transforms the physical subspace, $\mathbf{u}$, to the modal subspace, $\mathbf{p}$, and is defined by the following equation.

$$\begin{bmatrix} \mathbf{u}_c \\ \mathbf{u}_i \end{bmatrix} = \begin{bmatrix} \mathbf{g}_{cc_d} & \mathbf{Q}_{cl} \\ \mathbf{g}_{ic_d} & \mathbf{Q}_{il} \end{bmatrix} \begin{bmatrix} \mathbf{p}_c \\ \mathbf{p}_l \end{bmatrix} \tag{35}$$

In order to provide physical interface degrees of freedom, for use in component coupling, $\mathbf{p}_c$ in the above equation is back-transformed to eliminate it from the right-hand side of the equation. This results in the following transformation matrix,

$$\begin{bmatrix} \mathbf{u}_c \\ \mathbf{u}_i \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{cc} & \mathbf{0} \\ \mathbf{g}_{ic}{}^* & \mathbf{Q}_{il}{}^* \end{bmatrix} \begin{bmatrix} \mathbf{u}_c \\ \mathbf{p}_l \end{bmatrix} \tag{36}$$

where

$$\mathbf{g}_{ic}{}^* = \mathbf{g}_{ic_d} \mathbf{g}_{cc_d}{}^{-1} \tag{37}$$

$$\mathbf{Q}_{il}{}^* = \mathbf{Q}_{il} - \mathbf{g}_{ic_d} \mathbf{g}_{cc_d}{}^{-1} \mathbf{Q}_{cl} \tag{38}$$

The transformation of the component mass and stiffness matrices then proceeds in a similar manner as shown in Eqs. (14) to (23), with the following differences. The $\mathbf{Q}_{il}{}^*$ matrix partition replaces the $\mathbf{Q}_l$ matrix partition. The $\mathbf{g}_{ic}{}^*$ matrix partition replaces the Φ_{ic} matrix partition. In the fixed-interface methodology, the definition of the constraint modes, Φ_{ic}, leads to terms in the component stiffness matrix which cancel out. In the free-interface methodology, the definition of the transformation submatrices has changed and so this cancellation does not occur. Therefore, Eqs. (21) and (22) are replaced by the following equations.

$$\kappa_{cc} = \mathbf{g}_{ic}{}^{*T} (\mathbf{k}_{ii} \mathbf{g}_{ic}{}^* + \mathbf{k}_{ic}) + \mathbf{k}_{ci} \mathbf{g}_{ic}{}^* + \mathbf{k}_{cc} \tag{39}$$

and

$$\kappa_{jc} = \kappa_{cj}{}^T = \mathbf{Q}_{il}^{*\,T} (\mathbf{k}_{ii} \mathbf{g}_{ic}{}^* + \mathbf{k}_{ic}) \tag{40}$$

Free Interface Methodology for Components with Rigid Body Modes

When a component has rigid body modes, the associated stiffness matrix is singular. The inverse of the stiffness matrix, the flexibility matrix, cannot be directly obtained, and therefore the attachment modes cannot be directly obtained. To circumvent this problem, Rubin [11] presented the following method for obtaining the residual elastic attachment modes of a component with rigid body modes.

First, the stiffness matrix is constrained from rigid body motion by partitioning out r degrees of freedom, where r is the number of rigid body modes.

$$\mathbf{k} = \begin{bmatrix} \mathbf{k}_{ww} & \mathbf{k}_{wr} \\ \mathbf{k}_{rw} & \mathbf{k}_{rr} \end{bmatrix} \tag{41}$$

The remaining partition is then inverted.

$$\mathbf{g}_{ww} = \mathbf{k}_{ww}{}^{-1} \tag{42}$$

This flexibility matrix is then expanded back to n (w + r) size.

$$\mathbf{g}_c = \begin{bmatrix} \mathbf{g}_{ww} & \mathbf{0}_{wr} \\ \mathbf{0}_{rw} & \mathbf{0}_{rr} \end{bmatrix} \tag{43}$$

A square projection matrix is defined by

$$\mathbf{A} = \mathbf{I}_{nn} - \mathbf{m}\,\Phi_r \Phi_r^T \tag{44}$$

where Φ_r is the rigid body modes matrix. The elastic flexibility matrix, g_e, with rigid body motion removed, is shown in reference [11] to be

$$\mathbf{g}_e = \mathbf{A}^T \mathbf{g}_c\, \mathbf{A} \tag{45}$$

Now the analysis proceeds in a similar fashion to the previously discussed methodology of the free interface component with no rigid body motion. The major difference between the two approaches is that the elastic flexibility matrix is used in place of the general flexibility matrix. The inertia relief attachment modes are

$$\mathbf{g}_{a_e} = \begin{bmatrix} \mathbf{g}_{cc_e} \\ \mathbf{g}_{ic_e} \end{bmatrix} \tag{46}$$

The initial block of vectors is calculated using the inertia relief attachment modes and the elastic flexibility matrix.

$$\mathbf{q}_1^{**} = \mathbf{g}_e\, \mathbf{m}\, \mathbf{g}_{a_e} \tag{47}$$

The subsequent block-Krylov Ritz vectors are calculated, orthogonalized, and normalized as shown in Eqs. (28) to (31). The residual elastic flexibility terms are also calculated as shown in the free interface with no rigid body modes discussion, Eqs. (32) to (34). Creation of the transformation matrix, Eqs. (35) to (40), is also similar to when no rigid body modes are present. The one exception is that the rigid body modes must be included in the transformation matrix. Therefore, Eq. (35) is replaced by

$$\begin{bmatrix} \mathbf{u}_c \\ \mathbf{u}_i \end{bmatrix} = \begin{bmatrix} \mathbf{g}_{cc_d} & \mathbf{Q}_{cl} & \Phi_{cr} \\ \mathbf{g}_{ic_d} & \mathbf{Q}_{il} & \Phi_{ir} \end{bmatrix} \begin{bmatrix} \mathbf{p}_c \\ \mathbf{p}_l \\ \mathbf{p}_r \end{bmatrix} \tag{48}$$

Formation of the final transformation matrix, and subsequently the component mass and stiffness matrices, is then performed as described in the previous section.

Discussion

As previously discussed, most component mode synthesis applications use the normal eigenvalues of the substructure to form the component. It is accepted that if all the eigenvalues of a system are used to form the component, an "exact" finite element solution may be obtained. The same fact holds true for components based upon block-Krylov subspaces. Mathematically, this is proven in references [3] and [12]. If n orthogonal block-Krylov Ritz vectors are used to form a component of a n-size system, the same "exact" solution as from normal eigenvectors is obtained.

Orthogonality

Upon completion of the block-Krylov vector calculations and Gram-Schmidt orthogonalizations, all vectors are re-checked for orthogonality. Orthogonality checking is required because the normalized Gram-Schmidt orthogonalization algorithm is not always successful at producing independent vectors [6]. The Gram-Schmidt procedure will fail on occasions when vectors, although theoretically independent, are dependent within the numerical constraints of current digital computers. These vectors cannot then be made orthogonal and independent, using the normalized Gram-Schmidt procedure described in the previous sections.

The orthogonality of the block-Krylov vectors is re-checked using the following

$$\mathbf{L} = \mathbf{Q}_l^{\mathrm{T}} \mathbf{m}_{ii} \mathbf{Q}_l \tag{49}$$

If all vectors are orthogonal and normalized with respect to the mass matrix, **L** will be a l-size identity matrix. The mass matrix used in Eq. (49) is appropriate for fixed interface modes. For the free interface approach, the full physical mass matrix is used.

Several solutions to the numerical dependence problem are being investigated. These potential solutions include selective re-orthogonalization, the modified Gram-Schmidt procedure [6], and Kahan's orthogonalization procedure [13].

During examination of the boundary flexibility method, it was found that the Ritz vectors within the Krylov blocks were dependent on each other. In fact, it was determined that there is no theoretical basis why the vectors within the initial block should be independent. The boundary flexibility algorithm, as presented in reference [2] and discussed in section two, makes no orthgonality check of the vectors within the Krylov block. Furthermore, the subsequent Gram-Schmidt procedure is ineffective because the blocks are orthogonalized with respect to a set of vectors that are not orthogonal.

The problem of dependent vectors within the block can be eliminated by modifying the boundary flexibility algorithm. In order to make all the vectors in the solution orthogonal, each Krylov block was partitioned into it's individual vectors. The Gram-Schmidt orthonormalizing procedure, shown in Eqs. (9) to (12), was then applied vector by vector to each Krylov block. The constituent vectors are then reassembled and the analysis proceeds as before. An alternate solution to the problem of dependent vectors within the blocks, is shown in reference [12]. In this solution a singular value decomposition is performed on the $\mathbf{q}_j^{**\mathrm{T}} \mathbf{m} \mathbf{q}_j^{**}$ subspace. The resulting similarity transformation orthogonalizes the block.

Modal Selection

A subject which requires future investigation is the selection of block-Krylov component modes. The Krylov vectors are modes, although they are static modes (as are constraint and attachment modes), in contrast to the normal modes based upon the eigenvalues. The simplest approach to modal selection is modal truncation, based upon the numerical value of the natural frequency. However, modal truncation cannot be used directly with Krylov modes, because there are no eigenvalues associated with them. Modal selection techniques more sophisticated than modal truncation will also be investigated. The algorithm currently implemented has no modal selection capability. The Krylov modes produced are the Krylov modes used, in component formation.

Example

Programming

The previously described algorithms were implemented in MSC/NASTRAN [14]. The use of a standard, commercially available computer program allows the results of this work to be easily used by other structural dynamists. Adding these methodologies to MSC/NASTRAN is allowed through the use of the internal programming language called DMAP (Direct Matrix Abstraction Programming). The standard solution sequences of MSC/NASTRAN are written in DMAP, and the source code of MSC/NASTRAN is available at the DMAP level. For example, Eqs. (3) to (5), and Eqs. (16) to (23) are currently contained in the standard MSC/NASTRAN normal modes solution. Equations (6) to (12) were coded using DMAP, and were then incorporated into the MSC/NASTRAN solution sequence. A DMAP listing of the present methodology is contained in the Appendix.

Definition

An example case of a cantilevered beam was derived from a finite element model of the Space Station Freedom photovoltaic array central mast. The length of the beam was 1179.9 in. The modulus of elasticity, E, was 10.1×10^6 lb/in.^{2} and the moment of inertia of the cross section was 108.9 in.^{4} Its weight per unit length was 0.2296 lb/in. The cantilevered beam was modeled with eleven nodes and 10 beam finite elements. A variety of boundary conditions, described in the next section, were imposed upon this beam.

Results

A component representation of the 10 element beam was created using the boundary flexibility method with block-Krylov iteration. The fixed interface approach, with two Krylov blocks and constraint modes, was used to form the component. The interface of the component consisted of one node and 6 degrees of freedom. The number of constraint nodes is equal to the number of interface degrees of freedom, and the size of the Krylov block is equal to the number of constraint modes. Therefore, each Krylov block contained six vectors. Since the component was formed with two Krylov blocks, it contained a total of 12 generalized coordinates.

Plots of the lateral Krylov vectors, which represented the cantilevered beam, are shown in Figs. 1 to 4. The unorthogonalized vectors, as output by Eqs. (6) and (8) are shown in Figs. 1 to 2(a) and Figs. 3 to 4(a), respectively. The first normalized vector, as output by Eq. (7), is given in Fig. 1(b). (The first vector does not need to be orthogonalized.) The remaining orthogonalized and normalized vectors, as output by Eq. (12), are given in Figs. 2 to 4(b). The first two unorthogonalized Ritz vectors (plotted in Figs. 1 to 2(a)), which are in the first Krylov block, appear to be nearly identical. The first mode is similar to the classic first bending normal mode shape of a beam and, after Gram-Schmidt orthogonalization, the second vector has become the classic second bending normal mode shape (shown in Figs. 1 to 2(b)).

The eigenvalues of the reduced Krylov subspace were calculated next. The first five natural frequencies from this reduced system are shown in Table 1. For comparison, Table 1 also includes the frequencies of a reduced system where the component was formed using traditional normal modes. This component was also formed with a fixed interface, but thirteen normal modes were used for numerical convenience. The full, or "exact," finite element eigenvalue solution is also shown. In the case of the Krylov vectors, no modal selection of any kind was used. For the case of the normal modal component, modal selection by truncation was used. The superior accuracy of the normal modal component, in the fourth bending mode, does not necessarily represent a limitation of the block-Krylov method, but instead demonstrates the need for a Krylov modal selection criteria.

In addition to the fixed interface example, two free interface examples were created. Both were was based on the same 10 element beam described above, but with different boundary conditions. The first model was free-fixed, with the component interface being the free boundary condition, and hence it had no rigid-body modes. Equations (24) to (40) define the formulation of this free interface component. Table 2 shows the first five system natural frequencies of this model compared to the natural frequencies of the full finite element model. The other model consisted of the 10 element beam with free-free boundary conditions and rigid body modes. Equations (41) to (48) define the formulation of this component. Table 3 shows the first four elastic frequencies of this model compared to the frequencies of the full finite element model. In the two free interface cases, there is no comparison with the normal component mode synthesis. This is because standard free-interface MSC/NASTRAN routine does not use the "Rubin-MacNeal" method, and so a direct comparison was not performed.

Conclusions

Using boundary flexibility matrices to initialize the block-Krylov recurrence algorithm provides an efficient and simple method for generating static Ritz vectors. Static Ritz vectors so generated accurately represent the dynamics of a substructure. Because this methodology does not require the solution of the component eigenvalue problem, the component can be formed with a significant decrease in computational effort. Although the issues of vector dependency and modal selection require further investigation, the block-Krylov Component Mode Synthesis method is a promising alternative in dynamic substructuring.

References

[1] Craig, R.R.: Structural Dynamics, An Introduction to Computer Methods. Wiley, New York, 1981.

[2] Abdallah, A.A.; and Hucklebridge, A.A.: Boundary Flexibility Method of Component Mode Synthesis Using Static Ritz Vectors. Comput. Struct., vol. 35, no. 1, 1990, pp. 51–61.

[3] Craig, R.R.; and Hale, A.L.: Block-Krylov Component Synthesis Method for Structural Model Reduction. J. Guid. Control, Dyn., vol. 11, no. 6, 1988, pp. 562–570.

[4] Yiu, Y.C.; and Landess, J.D.: A Reduced Order Method for Passive Substructures. AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, 31st, Proceedings, Pt. 4, AIAA, Washington, DC, 1990, pp. 2036–2044.

[5] Nour-Omid, B.; and Clough R.W.: Dynamic Analysis of Structures Using Lanczos Co-Ordinates. Earthquake Eng. Struct. Dyn., vol. 12, 1984, pp. 565-577. (Also Report PAM–186, Center for Pure and Applied Mathematics, California Univ., Berkley, CA, 1984.)

[6] Noble, B.N.; and Daniel, J.W.: Applied Linear Algebra. Prentice-Hall, Englewood Cliffs, N.J., 1988.

[7] Wilson, E.L.; Yuan, M.W.; and Dickens, J.M.: Dynamic Analysis by Direct Superposition of Ritz Vectors. Earthquake Eng. Struct. Dyn., vol. 10, 1982, pp. 813–821.

[8] Wilson, E.L.; and Bayo, E.P.: Use of Special Ritz Vectors in Dynamic Substructure Analysis. J. Struct. Eng., Am. Soc. Civ. Eng., vol. 112, no. 8, 1986, pp. 1944–1954.

[9] Craig, R.R.; and Bampton, M.C.C.: Coupling of Substructures for Dynamic Analysis. AIAA J., vol. 6, 1968, pp. 1313–1319.

[10] MacNeal, R.H.: A Hybrid Method of Component Mode Synthesis. Comput. Struct., vol. 1, 1971, pp. 581–601.

[11] Rubin S.: Improved Component-Mode Representation for Structural Dynamic Analysis. AIAA J., vol. 13, no. 8, 1975, pp. 995–1006.

[12] Su, T.J.; and Craig R.R.: Model Reduction and Control of Flexible Structures Using Krylov Vectors. J. Guid. Control Dyn., vol. 14, no. 2, 1991, pp. 260–267.

[13] Parlett, B.N.: The Symmetric Eigenvalue Problem. Prentice-Hall, Englewood Cliffs, NJ, 1980.

[14] MSC/NASTRAN User's Manual, Version 67, The MacNeal-Schwendler Corporation, Los Angeles, CA, 1991.

TABLE 1.—COMPARISON BETWEEN THE CANTILEVERED BEAM FREQUENCIES USING FIXED INTERFACE COMPONENT NORMAL MODES, KRYLOV MODES, AND FINITE ELEMENTS

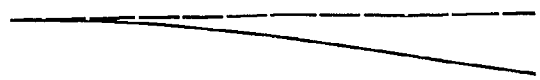
System frequencies full finite element		From component normal modes, Δ%		From component Krylov modes, Δ%	
1st Bend	0.5464 Hz	0.5464 Hz	0.	0.5464 Hz	0.
2nd Bend	3.415 Hz	3.415 Hz	0.	3.415 Hz	0.
3rd Bend	9.522 Hz	9.522 Hz	0.	9.610 Hz	.924
4th Bend	18.57 Hz	18.57 Hz	0.	24.24 Hz	30.5
1st Tors	27.58 Hz	27.58 Hz	0.	27.58 Hz	0.

TABLE 2.—COMPARISON BETWEEN THE CANTILEVERED BEAM FREQUENCIES USING FREE INTERFACE COMPONENT NORMAL MODES, KRYLOV MODES, AND FINITE ELEMENTS

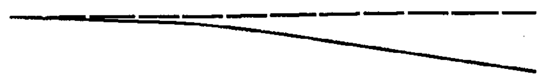
System frequencies full finite element		From component Krylov modes, Δ%	
1st Bend	0.5464 Hz	0.5464 Hz	0.
2nd Bend	3.415 Hz	3.415 Hz	0.
3rd Bend	9.522 Hz	9.524 Hz	.02
4th Bend	18.57 Hz	19.23 Hz	3.6
1st Tors	27.58 Hz	27.58 Hz	0.

TABLE 3.—COMPARISON BETWEEN THE FREE-FREE BEAM FREQUENCIES USING FREE INTERFACE COMPONENT KRYLOV MODES, AND FINITE ELEMENTS

System frequencies full finite element		From component Krylov modes, Δ%	
Rigid Body (6)	0. Hz	0.000 Hz	0.
1st Bend	3.474 Hz	3.474 Hz	0.
2nd Bend	9.549 Hz	9.549 Hz	0.
3rd Bend	18.65 Hz	18.69 Hz	.21
4th Bend	30.70 Hz	33.15 Hz	8.0

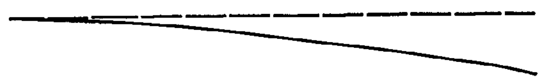


(a) Initial calculation.

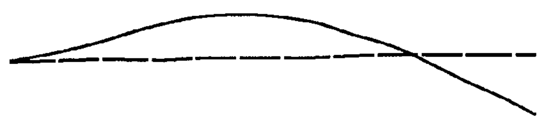


(b) After normalization.

Figure 1.—Fixed interface Krylov mode, block one, vector one.

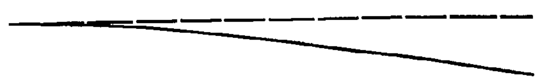


(a) Initial calculation.

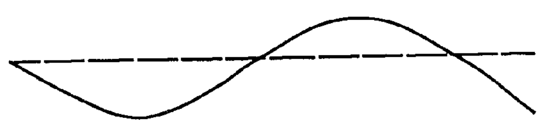


(b) After Gram-Schmidt orthogonalization.

Figure 2.—Fixed interface Krylov mode, block one, vector two.

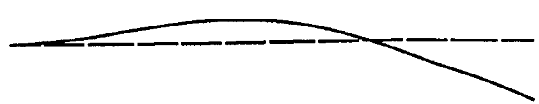


(a) Initial calculation.

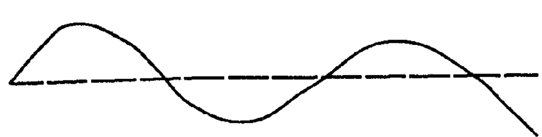


(b) After Gram-Schmidt orthogonalization.

Figure 3.—Fixed interface Krylov mode, block two, vector one.



(a) Initial calculation.



(b) After Gram-Schmidt orthogonalization.

Figure 4.—Fixed interface Krylov mode, block one, vector two.

Appendix - MSC/NASTRAN DMAP Listing of the Boundary

Flexibility Method of Block-Krylov Component Synthesis

```
$
$ BOUNDARY FLEXIBILITY METHOD DMAP IMPLEMENTATION
$
$  VERSION 1.0 - KELLY CARNEY
$          IMPLEMENTED ABDALLAH'S WORK AS PUBLISHED
$
$ NOTES ON USAGE
$
$ - A DUMMY METHOD AND EIGR CARD MUST BE USED TO INITIATE RITZ
$   VECTOR CALCULATION.
$
$ - MODAL DEGREES OF FREEDOM MUST BE PROVIDED AND SPECIFIED
$   ON SEQSET CARDS, AS WITH STANDARD NASTRAN SOLUTION
$
$ - MIXED BOUNDARY CONDITIONS (B AND C SET POINTS) ARE NOT YET
$   IMPLEMENTED.  USE B OR C, NOT BOTH.
$
$ - NOT COMPATIBLE WITH DYNAMIC REDUCTION, IF REQUESTED IT WILL
$   NOT BE PERFORMED
$
$ - AUTO-OMIT FEATURE HAS BEEN DISABLED. IF AUTOSPC IS ON,
$   AUTO-OMIT IS NOT REQUIRED
$
$ PARAMETERS
$
$  PARAM,BFLEX,-1 -  A NEGATIVE VALUE ON THIS PARAMETER TURNS ON THE
$            BOUNDARY FLEXIBILITY METHOD.
$            IF MULTIPLE SUPERELEMENTS ARE BEING PROCESSED,
$            PLACE IN SUBCASE SECTION OF CASE CONTROL
$
$  PARAM,L,I -     L IS THE NUMBER OF RITZ VECTOR DESIRED.  IT IS
$            SET BY THE NUMERICAL VALUE OF I
$
$  PARAM,DIAG,1 -   A POSITIVE VALUE CAUSES A LARGE AMOUNT OF MOSTLY
$            INDESCIPHERABLE DATA TO BE PRINTED
$
$ MATRICES
$
$  INPUT - CMKXX, CMMXX, GOAT, USET
$
$  OUTPUT,FIXED INTERFACE - GOQ,
$  OUTPUT,FREE INTERFACE - GOQ, REVISED MAA AND GOAT
$
$
$ SETUP BOUNDARY FLEXIBILITY PARAMETER AND DISABLE DYNAMIC
$ REDUCTION
$
ALTER   726 $
TYPE    PARM,,I,Y,(BFLEX=1) $
IF( BFLEX < 0 ) NODYNRED = -1 $
$
$ DISABLE AUTO-OMIT REDUCTION
$
ALTER   757 $
IF( BFLEX < 0 ) NOARED = -1 $
$
$ MOVE TO BOUNDARY FLEXIBILITY SECTION IF REQUESTED
$
ALTER   775 $
COND    BOUNFLX,BFLEX $  JUMP TO BOUNDRY FLEXIBILITY DMAP SECTION
$
ALTER   783 $
JUMP    ENDBF $
$
$ BEGIN BOUNDARY FLEXIBILITY METHOD SECTION
$
LABEL   BOUNFLX $
$
$ CREATE AND SET INITAL PARAMETERS
$
  TYPE     PARM,,I,N,(I=2) $         INCREMENT OF RITZ VECTORS
  TYPE     PARM,,I,N,(I1=2) $        INC OF RITZ VEC IN FIRST SET
  TYPE     PARM,,I,N,(J=2) $         INC OF RITZ VEC IN ITH SET
  TYPE     PARM,,I,N,(IR=2) $         INC OF R-SET
  TYPE     PARM,,I,N,(JSET) $         SIZE OF RITZ VEC SET
```

```
  TYPE    PARM,,RS,N,(MAXRAT) $      MAXRAT FROM DECOMP OF CMKXX
  TYPE    PARM,,I,Y,(L=1) $        TOTAL NUMBER OF RITZ VEC
  TYPE    PARM,,I,Y,(DIAG=-1) $      DIAGONOSTIC PRINTOUTS
  FILE    QP=APPEND,OVRWRT $        ALLOW QP TO APPEND,OVERWRITE
  FILE    QL=APPEND $             ALLOW QL TO BE APPENDED TO
  FILE    QNEW=APPEND $            ALLOW QNEW TO BE APPENDED TO
  FILE    PHIVZ=OVRWRT $           ALLOW PHIVZ TO BE OVERWRITTN
  FILE    NPHIVR=APPEND $          ALLOW NPHIVR TO APPEND
  FILE    MAA=OVRWRT $             ALLOW MAA TO BE OVERWRITTEN
  FILE    KAA=OVRWRT $             ALLOW KAA TO BE OVERWRITTEN
$
$ CHECK TO SEE IF BOTH B AND RC HAVE BEEN DEFINED
$
  IF(NORC>0 AND NOBSET>0) MESSAGE /
  /'FATAL ERROR - BOTH BSET AND CSET OR RSET DEGREES OF FREEDOM'/
  'HAVE BEEN DEFINED, ONLY ONE TYPE IS CURRENTLY ALLOWED'/ $
  IF(NORC>0 AND NOBSET>0) JUMP  RFERR $
$
$ PARTITION OUT SUPORT DEGREES OF FREEDOM, IF REQUESTED
$
  VEC     USET/VVRO/'V'/'R'/'COMP'/ $  CREATE PARTN VECTOR R-SET
  IF(NORSET >0) THEN $            EXECUTE ONLY IF R-SET
  PARTN   CMKXX,VVRO,/,,,KZZ/-1/////6 $ PARTION STIFFNESS MATRIX
$
  ELSE $                    IF NO R-SET
  EQUIV   CMKXX,KZZ/ALWAYS $        RENAME CMKXX TO KZZ
  ENDIF $
$
$ CALCULATE INITIAL RITZ VECTORS MATRIX
$
  DECOMP  KZZ/LZZ,/1/////S,N,SING/S,N,NBRCHG/
        S,N,MAXRAT/ $              DECOMP STIFFNESS MAT
  IF (SING=-1 OR NBRCHG>0 OR MAXRAT>1.E5) MESSAGE /
   /'FATAL ERROR-STIFFNESS SINGULAR-SUPPORT DOFS MAY BE REQUIRED'/$
  IF (SING=-1 OR NBRCHG>0 OR MAXRAT>1.E5) JUMP RFERR $ SINGULAR MAT
  DIAGONAL  KZZ/IZZ/'SQUARE'/0. $        Z-SIZE IDENTITY MAT
  FBS     LZZ,,IZZ/KINV/ $            CALCULATE K**-1
$
$    FOR FIXED BC METHOD
$
    IF(NOBSET>0) THEN $
    MPYAD   KINV,CMMXX,/KINVM/ $      CALCULATE K**-1*M
    MPYAD   KINV,MOA,/KINVMOA/ $      CALCULATE K**-1*MOA
    MPYAD   KINVM,GOAT,KINVMOA/Q1S/ $   CALCULATE INITIAL VEC
    UPARTN  USET,Q1S/Q1SA,,,/'A'/'B'/'Q'/2 $ PARTN DOWN TO B-SIZE
    ENDIF $
$
$    THIS SECTION IS EXECUTED FOR FREE INTERFACE METHOD
$
    IF(NORC>0) THEN $              IF EITHER R OR C SET
$
$    THIS SECTION IS EXECUTED FOR FREE INTERFACE METHOD,
$       WITH RIGID BODY MODES
$
    IF(NORSET>0) THEN $             ONLY IF R SET
$
$     CREATE AND ORTHO-NORMALIZE RIGID BODY TRANSFORMATION MATRIX
$
     VECPLOT, ,BGPDTS,EQEXINS,CSTMS,,USET/PHIGR/GRDPNT//6 $
     UPARTN   USET,PHIGR/PHIVR,,,/'G'/'V'/'S'/1 $ RIGID BODY MAT
$
$     EXTRACT FIRST VECTOR FROM MATRIX AND NORMALIZE
$
     MATMOD  PHIVR,,,,,/PHIVR1,/1/1 $  EXTRACT 1ST COLUMN
     SMPYAD  PHIVR1,CMMXX,PHIVR1,,,/NPH1/3////1////1$
     DIAGONAL  NPH1/NPH1S/'SQUARE'/-.5 $  SCALE FACTOR
     MPYAD   PHIVR1,NPH1S,/NPHIVR $     NORMALIZE VECTOR
$
$      ORTHOGONALIZE REMAINING RIGID BODY VECTORS
$
      DO WHILE (IR <= NORSET) $
$
      MATMOD   PHIVR,,,,,/PHIIR,/1/IR $    EXTRACT IRTH COLUMN
$
$      ORTHOGONALIZE IR VECTOR WITH PREVIOUS VECTORS
$
      SMPYAD  NPHIVR,CMMXX,PHIIR,,,/CIR/3////1 $ SCALE FACTOR
      MPYAD   NPHIVR,CIR,PHIIR/PHIIRS//-1 $ ORTHOGONALIZE
$
$      NORMALIZE ORTHOGONAL IR VECTOR
$
```

```
      SMPYAD   PHIIRS,CMMXX,PHIIRS,,,/NPHIIR/3////1////1$
      DIAGONAL  NPHIIR/NPHIIRS/'SQUARE'/-.5 $ SCALE FACTOR
      MPYAD    PHIIRS,NPHIIRS,/PHIIRSS $    NORMALIZE VECTOR
$
      APPEND   PHIIRSS,/NPHIVR/2 $        ADD NEW VECTOR
$
      IR = IR + 1 $                INCREMENT
      ENDDO $                      END DO LOOP
$
$     FORM INERTIA RELIEF MATRIX, GE (EQNS 21,22,23 - ABDALLAH)
$
     MERGE,   ,,,KINV,VVRO,/GC/-1//6 $    EXPAND KINV TO X-SIZE
     DIAGONAL  GC/IXX/'SQUARE'/0. $       CREATE IDENTITY MAT
     SMPYAD   CMMXX,NPHIVR,NPHIVR,IXX,,IXX/AXX/4/-1/1////1/ $
     SMPYAD   AXX,GC,AXX,,,/GE/3////1////6 $ CALCULATE GE
$
$     THIS SECTION IS EXECUTED IF FREE INTERFACE WITHOUT RIGID-BODY
$
     ELSE $
     EQUIV    KINV,GE/ALWAYS $         IF NO R-SET
     ENDIF $
$
$    FOR FREE INTERFACE, RIGID BODY OR NOT
$
    VEC     USET/VVCO/'V'/'C'/'COMP'/ $  C-SET PARTN MATRIX
    ADD     VVRO,VVCO/VVRCO///1 $       R AND C-SET PARTN MAT
    PARTN    GE,VVRCO,/GVRC,,,/1/ $     PARTITION OUT O COLS
    MPYAD    GE,CMMXX,/KINVM/ $         CALCULATE K**-1*M
    MPYAD    KINVM,GVRC,/Q1SA/ $        CALCULATE INITIAL VEC
    ENDIF $
$
$  EXTRACT FIRST VECTOR FROM MATRIX AND NORMALIZE FOR FIXED OR FREE
$
  MATMOD   Q1SA,,,,,/Q1PS,/1/1 $       EXTRACT 1ST COLUMN
  SMPYAD   Q1PS,CMMXX,Q1PS,,,/NQ1/3////1////1$
  DIAGONAL  NQ1/NQ1S/'SQUARE'/-.5 $    SCALE FACTOR
  MPYAD    Q1PS,NQ1S,/QP $           NORMALIZE VECTOR
$
$
COND PRINT1,DIAG $
MATPRN   KZZ,CMMXX,GOAT,PHIGR,PHIVR/ $
MATPRN   NPHIVR,GC,AXX,GE,KINVM/ $
MATPRN   GVRC,Q1SA,Q1PS,,/ $
PRTPARM  ///'NOBSET'/1 $
PRTPARM  ///'NOTSET'/1 $
PRTPARM  ///'NORC'/1 $
PRTPARM  ///'L'/1 $
LABEL PRINT1 $
$
$  ORTHOGONALIZE FIRST SET OF VECTORS
$
    DO WHILE (I1 <= NOTSET) $
$
    MATMOD   Q1SA,,,,,/Q1I1,/1/I1 $      EXTRACT I1TH COLUMN
$
$     ORTHOGONALIZE I1 VECTOR WITH PREVIOUS VECTORS
$
    SMPYAD   QP,CMMXX,Q1I1,,,/CI1/3////1 $ ORTHO SCALE FACTOR
    MPYAD    QP,CI1,Q1I1/Q1I1S//-1 $    ORTHOGONALIZE
$
$     NORMALIZE ORTHOGONAL I1 VECTOR
$
    SMPYAD   Q1I1S,CMMXX,Q1I1S,,,/NQ1I1/3////1////1$
    DIAGONAL  NQ1I1/NQ1I1S/'SQUARE'/-.5 $  SCALE FACTOR
    MPYAD    Q1I1S,NQ1I1S,/Q1I1SS $     NORMALIZE VECTOR
$
COND PRINT2,DIAG $
PRTPARM  ///'I1'/1$
MATPRN   QP,Q1I1,CI1,Q1I1S,Q1I1SS/ $
LABEL PRINT2 $
$
    APPEND   Q1I1SS,/QP/2 $          ADD NEW VECTOR
$         NOTE: IF NEW VECTOR IS NULL IT IS NOT APPENDED
$
    I1 = I1 + 1 $                INCREMENT
    ENDDO $                      END DO LOOP
$
$  INITIALIZE QL MATRIX
$
  COPY     QP/QL/ALWAYS/1 $         QL - SUM OF VECTORS
$
```

```
COND PRINT3,DIAG $
SMPYAD   QL,CMMXX,QL,,,/OCHK11/3////1 $
MATPRN   QL,OCHK11,,,/ $
LABEL PRINT3 $
$
$    BEGIN DO LOOP OF SOLVING FOR SUBSEQUENT VECTORS
$
     DO WHILE (I <= L) $
$
$    CALCULATION OF NEW SET OF VECTORS
$
     MPYAD    KINVM,QP,/QIS/ $       (K**-1)*RITZ PREVIOUS
     PARAML   QIS//'TRAILER'/1/S,N,NOJSET $ NO. COLUMS IN QIS
$
$    MAKE NEW SET OF VECTOR ORTHOGONAL TO PREVIOUS SETS
$
     SMPYAD   QL,CMMXX,QIS,,,/CI/3////1 $  CALCULATE SCALE
     MPYAD    QL,CI,QIS/QISS//-1 $        ORTHOGONALIZE
$
$    EXTRACT FIRST VECTOR FROM NEW SET AND NORMALIZE
$
     MATMOD   QISS,,,,,/QIJ1,/1/1 $       EXTRACT 1ST COLUMN
     SMPYAD   QIJ1,CMMXX,QIJ1,,,/NQI1/3////1////1$
     DIAGONAL NQI1/NQI1S/'SQUARE'/-.5 $   SCALE FACTOR
     MPYAD    QIJ1,NQI1S,/QNEW $          NORMALIZE VECTOR
$
COND PRINT4,DIAG $
PRTPARM  ///'I'/1/ $
MATPRN   QIS,CI,QISS,QIJ1,QNEW/ $
LABEL PRINT4 $
$
     J = 2 $                        RESET LOOP COUNTER
$
$       ORTHOGONALIZE NEW SET OF VECTORS WITH EACH OTHER
$
        DO WHILE (J <= NOJSET) $
$
        MATMOD   QISS,,,,,/QIJ,/1/J $    EXTRACT JTH COLUMN
$
$       ORTHOGONALIZE JTH VECTOR WITH PREVIOUS VECTORS
$
        SMPYAD   QNEW,CMMXX,QIJ,,,/CJ/3////1 $ ORTHO SCALING
        MPYAD    QNEW,CJ,QIJ/QIJS//-1 $  ORTHOGONALIZE
$
$       NORMALIZE ORTHOGONAL JTH VECTOR
$
        SMPYAD   QIJS,CMMXX,QIJS,,,/NQIJ/3////1////1$
        DIAGONAL NQIJ/NQIJS/'SQUARE'/-.5 $ SCALE FACTOR
        MPYAD    QIJS,NQIJS,/QIJSS $     NORMALIZE VECTOR
$
        APPEND   QIJSS,/QNEW/2 $         ADD NEW VECTOR
$
COND PRINT5,DIAG $
PRTPARM  ///'J'/1$
MATPRN   QIJ,CJ,QIJS,QIJSS,QNEW/ $
LABEL PRINT5 $
$
        J = J + 1 $                INCREMENT
        ENDDO $                     END DO LOOP
$
$    APPEND NEW VECTOR TO VECTOR MATRIX AND COMPLETE LOOP
$
     APPEND   QNEW,/QL/2 $      APPEND NEW VECTOR TO PREVIOUS
     COPY     QNEW/QP/ALWAYS/1 $ RESET PREVIOUS VECTOR TO NEW
$
     I = I + 1 $              INCREMENT
     ENDDO $                   END DO LOOP
$
$ FINISH PROCESSING MODE SHAPE MATRIX
$
   IF(NORSET>0) THEN $              IF R-SET, APPEND
   APPEND   NPHIVR,QL/CMPHIXZ/1 $       RIGID BODY MODES
   ELSE $                        IN NO R-SET, JUST
   EQUIV   QL,CMPHIXZ/ALWAYS $         CONVERT TO NASTRAN NAME
   ENDIF $
$
$ PERFORM MODE ORTHOGONALITY CHECK
$
   SMPYAD  CMPHIXZ,CMMXX,CMPHIXZ,,,/OCHKLL/3////1 $ PHI-T * M * PHI
   SMPYAD  CMPHIXZ,CMKXX,CMPHIXZ,,,/KBAR/3////1 $  PHI-T * K * PHI
$
```

```
  DIAGONAL OCHKLL/MI//-.5 $          DIAGONAL TERM AND ROOT
  DIAGONAL KBAR/KI//.5 $          DIAGONAL TERM AND ROOT
  ADD    MI,KI/Q/.159155//1 $       CALCULATE FREQUENCIES
  MATPRN  OCHKLL,KBAR,QL,Q,// $
$
PARAML  CMPHIXZ//'TRAILER'/1/S,N,NOZSET $  NUM RITZ = NUM EIGENVECTORS
$
LABEL   ENDBF $
$
$ TRANSFORMATION MATRIX CREATION
$  - BASED ON RUBIN-MACNEAL FOR FREE MODES (DOESN'T USE INREL)
$
ALTER 807 $
COND    TRANSF,BFLEX $          GO TO TRANSFORMATION DMAP
JUMP    ETRANSF $             JUMP AROUND IF NOT REQ.
$
ALTER 808 $
LABEL   TRANSF $             AVOID INREL MODULE
$
$ FREE BOUNDARY TRANSFORMATION
$
  IF(NORC>0) THEN $             EXECUTE ONLY IF RC SET
$
$ CALCULATE SYSTEM FLEXIBILITY WHEN RITZ VECTOR HAVE BEEN
$ ORTHOGONALIZED WRT THE MASS MATRIX (EQN. (15) ABDALLAH)
$
  SMPYAD   QL,CMKXX,QL,,,/GK1/3////1////6 $  INTER CALC
  SOLVE    GK1,/GK1INV/3 $             INVERT INTER CALC
  TRNSP    QL/QLT $                TRANSPOSE QL
  SMPYAD   QL,GK1INV,QLT,,,/GK/3////////6 $  CALC MODAL FLEX
$
$ CALCULATE RESIDUAL FLEXIBILITY MATRIX
$  NOTE: IF NO R-SET IS PRESENT THEN GE IS KINV
$
  ADD      GE,GK/GD//-1.0/0/ $          GD = GE - GK
$
$ PARTITION DOWN SQUARE FLEXIBILITY MATRIX TO C OR BOUNDARY DOFS
$
  PARTN    GD,VVRCO,/GCRCD,GORCD,,/-1//6/ $  GD = (GCRCD/GORCD)
$
$ FORM FREE "CONSTRAINT" MODE MATRIX, WHICH IS
$ THE LOWER LEFT PARTITION OF THE TRANSFORMATION MATRIX
$
  SOLVE    GCRCD,/GCCDINV/3 $           INVERT GCRCD
  MPYAD    GORCD,GCCDINV,/GOATFRS/ $       GOAT = GOCD*GCCD**-1
$
$ FORM FREE "MODAL" MATRIX, WHICH IS
$ THE LOWER RIGHT PARTITION OF THE TRANSFORMATION MATRIX
$
  PARTN    CMPHIXZ,,VVRCO/QRCL,QOL,,/1/ $    CMPHIXZ = (QRCL/QOL)
  MPYAD    GOATFRS,QRCL,QOL/PHIVZ//-1/ $     PHIVZ=QOL-GOAT*QRCL
$
$ PACK GOATFRS WITH ZEROS TO MAKE IT A-SIZE
$ THE LOWER LEFT PARTITION OF THE TRANSFORMATION MATRIX
$
  VEC      USET/VAQRC/'A'/'Q'/'COMP'/ $      FORM PART VECTOR
  MERGE,   ,,GOATFRS,,VAQRC,/GOATFREE/1/ $   MAKE A-SIZE
$
$ FORM REPLACEMENT BOUNDARY MASS AND STIFFNESS MATRIX, THIS
$ TRANSFORMATION IS THE SIMILAR TO STANDARD CMS AND SOL 63,
$ EXCEPT FOR FREE, REVISED 'GOAT' MATRIXIS NOT EQUAL TO
$ CONSTRAINT MODES, SO STIFFNESS TRNASFORMATION IS SAME AS MASS
$
  MPYAD MOO,GOATFREE,MOA/MOA1FR $          IDENTICAL TO 820
  MPYAD MOA,GOATFREE,MAA1/MAA2FR/1 $        INTERMEDIATE
  MPYAD GOATFREE,MOA1FR,MAA2FR/MAA/1////6 $   NEW MASS MATRIX
$
  MPYAD KOO,GOATFREE,KOA/KOA1FR $          STIFFNESS SECTION
  MPYAD KOA,GOATFREE,KAA1/KAA2FR/1 $        INTERMEDIATE
  MPYAD GOATFREE,KOA1FR,KAA2FR/KAA/1////6 $   NEW KAA MATRIX
$
$ REPLACE PREVIOUS GOAT WITH FREE ONE CALCULATED HERE
$
  EQUIV GOATFREE,GOAT/ALWAYS $
$
  ENDIF $
$
$ MAKE PHIVZ Q-SIZE, THIS REPLACES INREL FOR FREE AND FIXED
$ BOUNDARY FLEXIBILITY METHODS
$
IF (NOZSET > NOQSET) MESSAGE /
```

```
         /'FATAL ERROR-NUMBER OF MODES EXCEEDS Q-SET DOF'/$
IF (NOZSET > NOQSET) JUMP    RFERR $
$
MATGEN  ,/ZQ/6/NOQSET/NOZSET/NOQSET $ CREATE PARTITIONING MATRIX
MERGE   PHIVZ,,,,ZQ,/GOQ/1 $        TO MAKE PHIVZ Q-SIZE (GOQ)
$
COND    PRINT6,DIAG $
MATPRN  CMPHIXZ,GK1,GK1INV,GK,GD//$
MATPRN  GORCD,GORCD,GCCDINV,GOATFRS,QRCL//$
MATPRN  QOL,PHIVZ,GOATFREE,MAA,KAA// $
LABEL   PRINT6 $
$
LABEL   ETRANSF $
$
$ SETUP INTERMEDIATE MODAL STIFFNESS MATRIX
$
ALTER 816,816 $
UMERGE1  USET,KQQ,,,/KLAA1/'A'/'Q'/'T'/0 $
$
$ CREATE OFF-DIAGONAL MODAL STIFFNESS TERMS
$
ALTER 818 $
IF(NORC>0 AND BFLEX<0) THEN $
MPYAD   KOO,GOAT,KOA/KOA1 $
MPYAD   GOAQ,KOA1,/KQT/1 $
TRNSP   KQT/KTQ $
ADD     KQT,KTQ/KAAQT $
ADD     KLAA1,KAAQT/KLAA $
ELSE $
EQUIV   KLAA1,KLAA/ALWAYS $
ENDIF $
$
ALTER  859 $
COND   PRINT7,DIAG $
MATPRN  GOQ,KLAA,MLAA,,// $
LABEL  PRINT7 $
$
ENDALTER $
```